

Programación

Trabajo Práctico Individual

- 2) a. Leer la función **PerformMovement** del script `/Assets/Scripts/Mechanics/KinematicObject.cs` e identificar si se utilizan **colliders** o **raycasts** para detectar las superficies que el objeto tiene por delante.
- Tener en cuenta la variable **"hitBuffer"** y la llamada a la función **Cast** en la línea **135**.

La función `/Assets/Scripts/Mechanics/KinematicObject.cs` utiliza raycasts para detectar las superficies que el objeto tiene por delante. Esto se hace mediante la llamada a la función `"body.Cast(move, contactFilter, hitBuffer, distance + shellRadius)"` en la línea 135. Ésta línea 135, ejecuta un cast de un rayo desde la posición actual del objeto en la dirección especificada por el vector `"move"`. Los resultados de este cast se almacenan en el arreglo `"hitBuffer"`. Si hay colisiones con objetos en la dirección del movimiento, esos objetos se detectan y se almacenan en `"hitBuffer"`.

Luego, en un bucle `"for"`, se itera a través de las colisiones detectadas en `"hitBuffer"`. Se verifica la normal de la superficie en el que el objeto ha colisionado, y si esta normal cumple con ciertos criterios (por ejemplo, si es lo suficientemente vertical), se considera que el objeto está en tierra (`"IsGrounded = true"`). Además, se ajusta la velocidad del objeto para tener en cuenta la colisión con la superficie.

- b. ¿Qué tipo de estructura de datos se utiliza para almacenar los tokens (objetos coleccionables) para que puedan ser animados todos al mismo tiempo en el archivo `/Assets/Scripts/Mechanics/TokenController.cs`
- Tener en cuenta la variable **"tokens"** y que se puede acceder a **cualquiera de sus elementos** desde la función **Update()**

En el archivo `/Assets/Scripts/Mechanics/TokenController.cs`, la estructura de datos utilizada para almacenar los tokens es un array. La variable `"tokens"` es un array de elementos tipo `"TokenInstance"`.

En este caso, el array `"tokens"` se utiliza para mantener una colección de instancias de `"TokenInstance"`. Cada elemento del array representa un token individual que puede ser animado.

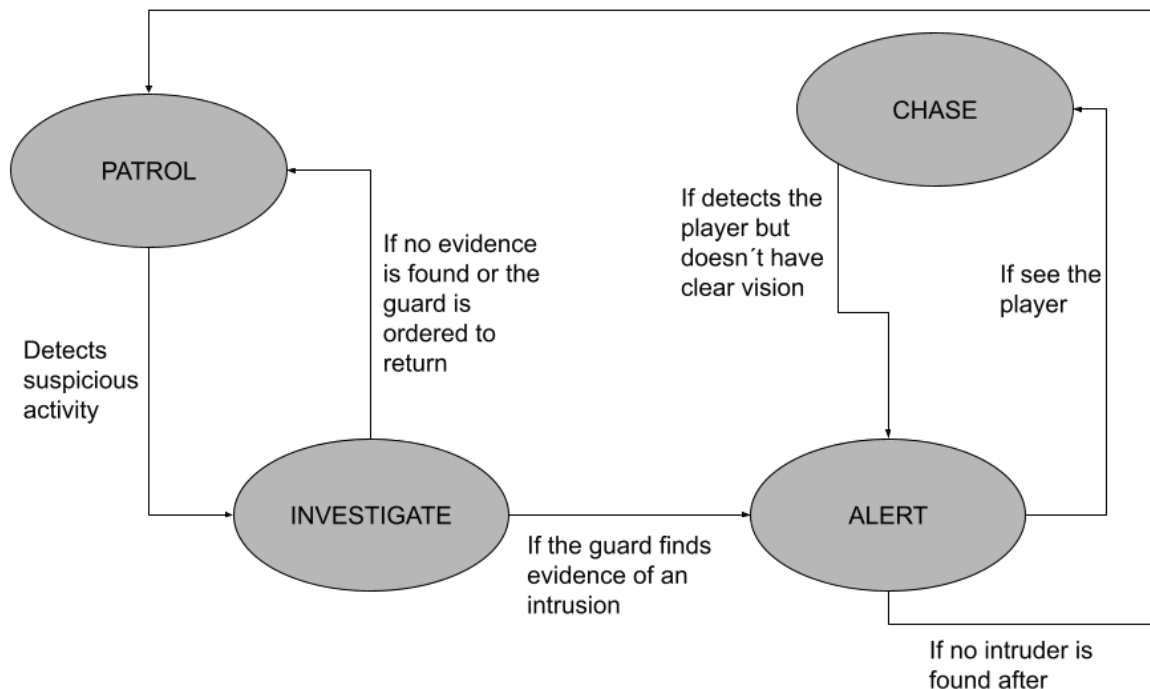
La ventaja de usar un array en este contexto es que permite acceder a cualquiera de los elementos desde la función `"Update()"`, lo que facilita la animación de todos los tokens al mismo tiempo. El bucle `"for"` en la función `"Update()"` itera a través de los elementos del array `"tokens"` y realiza las operaciones de animación en cada uno de ellos.

- c. Elegir un juego que hayas jugado recientemente y diagramar un árbol de comportamiento o máquina de estado de alguna de sus inteligencias artificiales.

Juego elegido: Hitman

El jugador asume un papel de un asesino a sueldo llamado Agent 47, y su objetivo es llevar a cabo asesinatos sigilosos en diversos entornos. Los NPCs en `"Hitman"` pueden variar en

tipo, pero para este ejemplo voy a considerar a los guardias de seguridad como el tipo de NPCs en el que basaré el diagrama de la máquina de estado.



d. Jugar el juego corriendo el ejecutable /Build/**Platformer.exe** y redactar los bugs encontrados y sus posibles causas.

Bugs encontrados y sus posibles causas:

1. Hay un enemigo que no ataca y el jugador al atacar, éste no cambia de estado, no muere:

Posible causa: podría deberse a un problema en la lógica del comportamiento del enemigo, como su máquina de estados o sus condiciones de ataque. También podría estar relacionado con la detección de colisiones entre el jugador y el enemigo.

En la función "update", la lógica que maneja los ataques y la detección de colisiones podría no estar funcionando correctamente.

2. Hay una plataforma sin colisión:

Posible causa: puede deberse a una configuración incorrecta del objeto de la plataforma en el juego. Puede ser que su componente "Collider2D" no esté configurado correctamente.

Habría que ver que esté configurado como un colisionador sólido y que las capas de colisión estén configuradas correctamente, además de verificar que la plataforma esté configurada como una plataforma que el jugador puede atravesar y usar para saltar.

3. Hay un diamante que no desaparece después de ser recogido:

Posible causa: este problema podría deberse a un error en el script que maneja la recolección de diamantes. Es posible que el script no esté eliminando el diamante de la escena después de que el jugador lo recoja o una falta de actualización adecuada de las estructuras de datos que almacenan el estado de los diamantes recogidos.